Journal of Ambient Intelligence and Smart Environments 0 (0) 1
IOS Press

Closure properties for fuzzy recursively enumerable languages and fuzzy recursive languages

Antonio Diego Silva Farias [a,*],
Luiz Ranyer de Araújo Lopes [b]
Benjamín Bedregal [b]
Regivan H. N. Santiago [b]

[a] *Federal Rural University of Semi-Arid - UFERSA, Universitary Campus of Pau dos Ferros, BR 226, KM 405, São Geraldo, 59.900-000, Pau dos Ferros, RN, Brazil*

[b] *Department of Informatics and Applied Mathematics - DIMAp, Federal University of Rio Grande do Norte - UFRN, Avenue Senador Salgado Filho, 3000, Universitary Campus, Lagoa Nova, 59.078-970, Natal, RN, Brazil*

Abstract. There are several variations of fuzzy Turing machines in the literature, many of these forms require a t-norm in order to establish their accepted language. This paper generalize the concept of non-deterministic fuzzy Turing machine - NTFM, replacing the t-norm operator for several aggregation functions, we establish the languages accepted by these machines, called fuzzy recursively enumerable languages or simply *LFRE* and show, among others results, which the class *LFRE* is closed to unions and intersections of fuzzy languages.

[*]Corresponding author. E-mail: antonio.diego@ufersa.edu.br

Keywords: `Turing machines, Fuzzy Turing machines, Fuzzy languages, Aggregation functions`

1. Introduction

The starting point for the computer science was to determine a model able to allow machines to be able to perform calculations previously performed only by humans. One of these computing models, called the Turing machines - TM, was proposed in 1936 by Alan Turing [28,29], and has since become the most accepted model by scientific community.

Lofti Zadeh, in [38], generalize the discrete notion, 0 or 1, of membership or non-membership in classical set theory for the continuous interval $[0, 1]$, resulting in the fuzzy sets theory. This enabled several extensions of classical theories were built from the idea of Zadeh [7], the fuzzy Turing machines [2,20,21,37] are an example of these generalizations.

The first formulation of fuzzy Turing machine was introduced by Zadeh himself (see [39]), at the end of the 1960s, through what he called fuzzy algorithm. Then Zadeh, along with Lee have defined the concept of fuzzy language [19]. After these papers, Santos [30,31], show that the fuzzy algorithms and fuzzy Turing machines are equivalent models.

For a long time there were no new works focused on fuzzy Turing machines, only with the work Harkleroad [16] have suggested researchers interested in studying this theme (see [5,6,8,11,12,24,25,26]). In recent years, mainly motivated by the possibility of extrapolating the Church-Turing thesis, there were several other approaches to computability (for example: [9,10,14,34]). Besides all this, Wiedermann [35,36] show that it is

[*]Corresponding author. E-mail: antonio.diego@ufersa.edu.br

possible to solve the halting problem, i.e., that it can accept recursively enumerable languages *r.e.* and also *co-r.e.*. With the work of Wiedermann, arose researchs as the Bedregal and Figueira [2], which proved not to be possible to define a universal fuzzy Turing machine.

In this paper we will discuss about non-deterministic fuzzy Turing machines - NTFM and their languages. We propose a new variation of NTFM, and study the properties of fuzzy languages that can be computed by these machines. But before starting the proposed work, we present some very important definitions, so that the reader has a good understanding of work.

We began the paper with basic notions of fuzzy sets, their origin and important definitions such as: union, intersection, complement and crisp sets, using the opportunity to also introduce the concepts of t-norms, t-conorms and fuzzy negation. Then we define the fuzzy languages and their basic operations. In Section 3, we define classical Turing machines, we present some examples, we define our variation of fuzzy Turing machines of its class languages, the fuzzy recursively enumerable languages. Finally, in Section 4, we study operations as union and intersection of fuzzy language in the class of these languages and proved that the class of fuzzy recursively enumerable languages is closed for these operations.

2. The fuzzy set

The fuzzy set theory was proposed by Lotfi A. Zadeh [38], in 1965, in order to generalize the discrete notions of membership and non-membership of classical set theory for the interval $[0, 1]$. According to Zadeh, some classes of objects in real-world physical can not always be precisely separated into distinct classes, such as: the “class of tall men" and the “class of non-tall men".

Formally, a **fuzzy set** A on a universe U is characterized by its membership function $\mu_A : U \to [0, 1]$, where for all $x \in U$ the value $\mu_A(x) \in [0, 1]$ represents the degree of membership of x in A.

The operations between the fuzzy sets are performed as follows:

Definition 1. *Let A and B two fuzzy sets on a universe U respectively represented by the memberships functions μ_A and μ_B. Then,*

1. *The* **union** *of A and B is the fuzzy set $A \cup B$ with membership function*

$$\mu_{A\cup B}(x) = max(\mu_A(x), \mu_B(x)).$$

2. *The* **intersection** *of A and B is the fuzzy set $A \cap B$ with membership function*

$$\mu_{A\cap B}(x) = min(\mu_A(x), \mu_B(x)).$$

3. *The* **complement** *of fuzzy set A is the fuzzy set A^C with membership function*

$$\mu_{A^C}(x) = 1 - \mu_A(x).$$

4. *The* **crisp** *of a classical set X is the fuzzy set with membership function*

$$\mu_X(x) = \begin{cases} 1, & se\ x \in X; \\ 0, & otherwise \end{cases}$$

2.1. Triangular norms and conorms

The first formulation for triangular norm, also called of t-norm, was given by Menger [23],who studied generalization of triangle inequality of metric spaces for probabilistic metric spaces. However, the most adopted at the present definition was given by Schweizer and Sklar in [32,33].

The t-norms and your dual operators, the t-conorms, provide mathematical models for the conjunctions and the disjuntions for fuzzy logic [18], respectively. The following are the definitions of these operators:

Definition 2. *A* **t-norm** *is a function $T : [0, 1]^2 \to [0, 1]$ which satisfies the following properties:*

(T1) $T(x, 1) = x$*, for any* $x \in [0, 1]$*;*
(T2) $T(x, y) = T(y, x)$*, for all* $x, y \in [0, 1]$*;*
(T3) $T(x, y) \leq T(x, z)$*, for any* $x, y, z \in [0, 1]$ *with* $y \leq z$*;*
(T4) $T(x, T(y, z)) = T(T(x, y), z)$*, for any* $x, y, z \in [0, 1]$.

Definition 3. *A* **t-conorm** *is a function $S : [0, 1]^2 \to [0, 1]$ which satisfies the following properties:*
(S1) $S(x, 0) = x$*, for any* $x \in [0, 1]$*;*
(S2) $S(x, y) = S(y, x)$*, for all* $x, y \in [0, 1]$*;*
(S3) $S(x, y) \leq S(x, z)$*, for any* $x, y, z \in [0, 1]$ *with* $y \leq z$*;*
(S4) $S(x, (S(y, z))) = S(S(x, y), z)$*, for any* $x, y, z \in [0, 1]$.

Both t-norms and t-conorms are examples of aggregation functions which difine below:

Definition 4. *An* **aggregation function** *(see [3]) is a function $A : [0; 1]^2 \to [0; 1]$, such that:*

(A1) $A(0,0) = 0$ *and* $A(1,1) = 1$*;*
(A2) *If* $x_1 \leq x_2$ *and* $y_1 \leq y_2$*, then* $A(x_1, x_2) \leq A(y_1, y_2)$*.*

To simplify the notation we use $a \circledast b$ *rather than* $A(a, b)$*.*

In table 1, we present some examples of t-norms and t-conormas.

Although t-norms and t-conormas be defined for two input arguments, the associative property of these functions allows us to define an extended form for these operators. Given a t-norm T and $(x_1, \cdots, x_n) \in [0,1]^n$ we denote by:

$$T_n(x_1, \cdots, x_n) = \begin{cases} 1, & \text{if } n = 0; \\ T(T_{n-1}(x_1, \cdots, x_{n-1}), x_n), & \text{if } n \geq 1 \end{cases}$$

In addition, t-norms can be extended for an operation with a countable number of inputs, writing:

$$T_\infty(x_1, x_2, \cdots) = \lim_{n \to \infty} T_n(x_1, \cdots, x_n),$$

where $(x_i)_{i \in \mathbb{N}}$ is a sequence of numbers that belong the interval $[0,1]$. Further details can be found at [18].

Analogously, we define the t-conorms extended to any countable number of inputs with only one exception: $S_0(\emptyset) = 0$.

As previously mentioned, the t-norms and t-conorms are generalized forms of conjunctions and disjunctions of classical logic, to fuzzy world. The negations can also be defined in this context:

Definition 5. *A* **fuzzy negation** *is a function* $N : [0,1] \to [0,1]$ *such that:*

(N1) $N(0) = 1$ *and* $N(1) = 0$*;*
(N2) $x \leq y \Rightarrow N(y) \leq N(x)$*.*

Example 1. *The functions:*

(i) $N_C(x) = 1 - x$*,*

(ii) $N_\perp(x) = \begin{cases} 1, & \text{if } x = 1 \\ 0, & \text{if } x \in (0,1] \end{cases}$*, and*

(iii) $N_\top(x) = \begin{cases} 1, & \text{if } x \in (0,1] \\ 0, & \text{if } x = 0 \end{cases}$

are examples of fuzzy negations.

For any aggregation function $A : [0,1]^2 \to [0,1]$ and any fuzzy negation N, we build a new operator $A_N(x,y) = N(A(N(x), N(y)))$ called of **dual operator** of A with respect to N. It is not difficult to show that the following proposition is true:

Proposition 1 ([18])**.** *Let* $T : [0,1]^2 \to [0,1]$ *a aggregation function and* $N : [0,1] \to [0,1]$ *a strong negation, i,e., with* $N(N(x)) = x$ *for all* $x \in [0,1]$*. Then,* T *is a t-norm if, and only if,* T_N *is a t-conorm.*

2.2. Fuzzy languages

The fuzzy languages was introduced in 1969 by Lee and Zadeh in [19], which also establish a series of valid operations on fuzzy languages such as: a union, intersection, complement and concatenation. formally we have:

Definition 6. (i) *A* **fuzzy language** *is a fuzzy set on a universe* V_T^**, where* V_T *represents a set of terminal objects and* V_T^* *denotes the set of all strings formed from elements of* V_T*. Thus, a fuzzy language is a set of the form*

$$\mathscr{L} = \{(x, \mu_{\mathscr{L}}(x)) : x \in V_T^*\}$$

(ii) *Given two fuzzy languages* $\mathscr{L}_1$ *e* $\mathscr{L}_2$ *on* V_T^**, the* **union** *of* $\mathscr{L}_1$ *and* $\mathscr{L}_2$ *is the fuzzy language defined by*

$$\mathscr{L}_1 \cup \mathscr{L}_2 = \{(x, \mu_{\mathscr{L}_1 \cup \mathscr{L}_2}(x)) : x \in V_T^*\},$$

where $\mu_{\mathscr{L}_1 \cup \mathscr{L}_2}(x) = max(\mu_{\mathscr{L}_1}(x), \mu_{\mathscr{L}_2}(x))$*.*

(iii) *The* **intersection** *of two fuzzy languages* $\mathscr{L}_1$ *and* $\mathscr{L}_2$ *on* V_T^* *is represented by*

$$\mathscr{L}_1 \cap \mathscr{L}_2 = \{(x, \mu_{\mathscr{L}_1 \cap \mathscr{L}_2}(x)) : x \in V_T^*\},$$

where $\mu_{\mathscr{L}_1 \cap \mathscr{L}_2}(x) = min(\mu_{\mathscr{L}_1}(x), \mu_{\mathscr{L}_2}(x))$*.*

(iv) *The fuzzy language* **dual** $\mathscr{L}$ *on* V_T^* *is:* $\mathscr{L}_d = \{(x, \mu_{\mathscr{L}_d}(x)) : x \in V_T^*\}$*, where*

$$\mu_{\mathscr{L}_d}(x) = \begin{cases} 0, & \text{if } \mu_{\mathscr{L}}(x) = 0 \\ 1 - \mu_{\mathscr{L}}(x), & \text{otherwise} \end{cases}$$

(v) *The* **reverse** *language of a language* $\mathscr{L}$ *is the fuzzy language given by:* $\mathscr{L}^R = \{(w^R, \mu_{\mathscr{L}(w)}) : w \in V_T^*\}$*, where* w^R *denote the string* w *written in reverse order.*

Rewark 1. *For notational simplicity, in the examples presented here, just present the support of fuzzy languages. For example,* $\mathscr{L} = \{(aa, 0.1), (ba, 0.8)\}$ *is the fuzzy language defined by the following membership function:*

$$\mu_{\mathscr{L}}(w) = \begin{cases} 0.1;, & \text{if } w = aa \\ 0.8, & \text{if } w = ba \\ 0, & \text{otherwise} \end{cases}$$

t-norms	**t-conorms**
$T_{mim}(x,y) = min(x,y)$	$S_{max}(x,y) = max(x,y)$
$T_P(x,y) = x \cdot y$	$S_P(x,y) = x + y - xy$
$T_{LK}(x,y) = max(x+y-1,0)$	$S_{LK}(x,y) = min(x+y,1)$
$T_D(x,y) = \begin{cases} 0, & \text{if } x,y \in [0,1) \\ min(x,y), & \text{otherwise} \end{cases}$	$S_D(x,y) = \begin{cases} 1, & \text{if } x,y \in (0,1] \\ max(x,y), & \text{otherwise} \end{cases}$

Table 1

Examples of t-norms and t-conorms

Example 2. *Let $V_T = \{0,1\}$. Clearly $\mathscr{L} = \{(0,0.5), (1,0.8), (00,1.0), (10,0.1), (01,1.0), (11,0.3)\}$ and $\mathscr{L}' = \{(0,0.8), (1,0.4), (00,0.7)\}$ are fuzzy languages, where all strings in $\{0,1\}^*$ not listed in $\mathscr{L}$ and $\mathscr{L}'$ have respective degrees of membership null. Furthermore,*

$$\mathscr{L} \cup \mathscr{L}' = \{(0,0.8), (1,0.8), (00,1.0), (10,0.1), (01,1.0), (11,0.3)\}$$

$$\mathscr{L} \cap \mathscr{L}' = \{(0,0.5), (1,0.4), (00,0.7)\}$$

$$\mathscr{L}_d = \{(0,0.2), (1,0.2), (10,0.9), (11,0.7)\}$$

$$\mathscr{L}^R = \{(0,0.5), (1,0.8), (00,1.0), (01,0.1), (10,1.0), (11,0.3)\}$$

These concepts can be generalized using t-norms, t-conorms and fuzzy negations. Given a t-norm T, a t-conorm S and a strong negation N, we define:

(i) The union with respect to S of fuzzy languages $\mathscr{L}$ and $\mathscr{L}'$ by:

$$\mathscr{L} \cup_S \mathscr{L}' = \{(w, S(\mu_{\mathscr{L}}(w), \mu_{\mathscr{L}'}(w))) : w \in V_T^*\}$$

(ii) The intersection with respect to T of fuzzy languages $\mathscr{L}$ and $\mathscr{L}'$ by:

$$\mathscr{L} \cap_T \mathscr{L}' = \{(w, T(\mu_{\mathscr{L}}(w), \mu_{\mathscr{L}'}(w))) : w \in V_T^*\}$$

(iii) The dual of language $\mathscr{L}$ with respect to fuzzy negation N by:

$$\mathscr{L}_N = \{(w, \mu_{\mathscr{L}_N}(w))) : w \in V_T^*\},$$

where $\mu_{\mathscr{L}_N}(w) = \begin{cases} N(\mu_{\mathscr{L}}(w)), & \text{if } \mu_{\mathscr{L}}(w) > 0 \\ 0, & \text{otherwise} \end{cases}$

3. Turing machines

3.1. Classical Turing machines

Turing machines consists of a computing model that describes a situation in which a person performs calculations on a paper tape using only two tools, one that lets you write a symbol on the sheet and another that allows you to erase the data on the tape. These calculations are performed through three basic operations that allow the person: (1) Read a particular symbol on the tape; (2) Change a symbol on the tape and (3) Move the read head by tape. In this sense, we can find different approaches, as example, [17,22,26].

Formaly a non-deterministic Turing machine - NTM is a septuple $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F \rangle$, where Q is a finite set of states, $\Sigma \subseteq \Gamma$ is a set of input symbols, Γ is a set of symbols allocable on the tape, $\Box \in \Gamma - \Sigma$ is a spetial character named blank, $q_0 \in Q$ is the initil state of machine, $F \subseteq Q$ is a set of final states and $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{R, L\}$ is a set of valid instructions.

In this paper, we consider that the NTM's have read tape with infinite length, however, left finite as we represent in figure 1.

Definition 7. *A* **valid instruction** *is a $l = (q_1, \sigma_1, q_2, \sigma_2, X) \in \delta$, where $q_1, q_2 \in Q$, $\sigma_1, \sigma_2 \in \Gamma$ e $X \in \{R, L\}$, describes the process of reading and writing of NTM, meaning that the NTM at a given moment was in the state q_1, performed the read symbol σ_1, moved to the state q_2 replacing the symbol σ_1 by σ_2 and moves the read head to the right or left (if $X = R$ or $X = L$, respectively).*

The above operations can be described by *instantaneous descriptions* (ID's), which consists in formulas of type uqv, with $q \in Q$ and $u, v \in \Gamma^*$[1]. The tape reading

[1] Γ^* denote the set of all finite strings about Γ including the empty string ε.

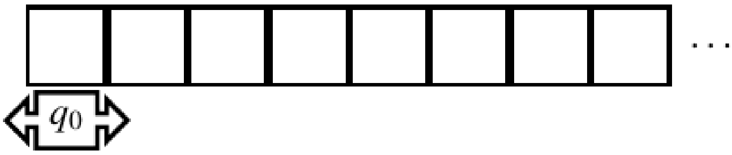

Fig. 1. Work tape of a Turing machine

is always done from left to right, reading the symbol from the most left v.

Given two ID's D_1 and D_2 of a NTM M, we say that there is a valid movement to D_1 from D_2 and written $D_1 \vDash D_2$ if one of the following cases can be verified:

Case 1: If $D_1 = qx_1x_2\cdots x_n$, with $x_i \in \Gamma$, then

$$D_2 = \begin{cases} ypx_2x_3\cdots x_n \text{ , if } (q, x_1, p, y, R) \in \delta \\ pyx_2x_3\cdots x_n \text{ , if } (q, x_1, p, y, L) \in \delta \end{cases}$$

Case 2: If $D_1 = x_1\cdots x_iqx_{i+1}\cdots x_n$, then

$$D_2 = \begin{cases} x_1\cdots x_iypx_{i+2}\cdots x_n \text{ , if } (q, x_{i+1}, p, y, R) \in \delta \\ x_1\cdots px_iyx_{i+2}\cdots x_n \text{ , if } (q, x_{i+1}, p, y, L) \in \delta \end{cases}$$

Case 3: If $D_1 = x_1x_2\cdots x_nq$, then

$$D_2 = \begin{cases} x_1x_2\cdots x_nyp \text{ , if } (q, \Box, p, y, R) \in \delta \\ x_1x_2\cdots px_ny \text{ , if } (q, \Box, p, y, L) \in \delta \end{cases}$$

We can define a more complete notion of movement on the tape of a NTM, extending $\vDash$ recursively.

Definition 8. *We say that $uqv \vDash^* u'q'v'$ if $uqv = u'q'v'$ or is there a ID wpz such that $uqv \vDash^* wpz \vDash u'q'v'$, in other words, $\vDash^*$ is the reflective and transitive closure of $\vDash$. Thus, the language accepted by NTM is the set:*

$$\mathcal{L} = \{w \in \Sigma^* : q_0w \vDash^* upv \text{ for } p \in F \text{ and } u, v \in \Gamma^*\}$$

Example 3. *Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, q_f\rangle$, where $Q = \{q_0, q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \Box\}$ and $\delta = \{(q_0, \Box, q_f, \Box, R), (q_0, 0, q_1, \Box, R), (q_1, 0, q_1, 0, R), (q_1, 1, q_1, 1, R), (q_1, \Box, q_2, \Box, L), (q_2, 1, q_3, \Box, L), (q_3, 0, q_3, 0, L), (q_3, 1, q_3, 1, L), (q_3, \Box, q_0, \Box, R)\}$. Then M is a NTM with accepted language $\mathcal{L} = \{0^{*n}1^{*n} : n \in \mathbb{N}\}$, where a^{*n} is the string formed by concatenating n symbols a. The figure 2 present a graphical representation of the Turing machine M.*

3.2. Fuzzy Turing machines

After of work to Zadeh [38], several extensions to classical theories have been developed to fuzzy world (see [7]), automata and Turing machines are examples these generalizations. The first models of fuzzy Turing machines were established by Zadeh [39], Lee [19] and Santos [30] which of course also studied the languages accepted by them.

Many variations of fuzzy Turing machines can be found in the literature, we can cite as examples [2,20, 21,35,36,37], in all these studies the use of a unique operator (as a t-norm, for example) is of paramount importance to establish the languages accepted by these machines. However, t-norms have characteristics that can be discarded, as commutativity and associativity. In this sense, we propose the following formulation of fuzzy Turing machine.

Definition 9. *A non-deterministic fuzzy Turing machine - NFTM is a septuple $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F, \{\circledast_q\}_{q\in Q-F}, \mu\rangle$, such that $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F\rangle$ is a NTM, for each $q \in Q - F$ we have one computable aggregation function[2] $\circledast_q : [0, 1]^2 \to [0, 1]$ associated and $\mu : \delta \to (0, 1]$ is a function which defines the degree of truth of one instruction in δ.*

Rewark 2. (1) *Note that in our definition, if $(q, u, q', v, E) \in \delta$ (com $E \in \{L, R\}$) is a instruction valid of NTM, then the degree of truth for this instruction needs to be strictly positive, i.e, $\mu(l) > 0$ for all instruction $l \in \delta$, otherwise does not occur transition. Thus, for any invalid instruction, i.e, with $l \notin \delta$, we have to $\mu(l) = 0$.*

(3) *For simplicity, we assume from now on that all functions occurring in this paper are computable in some sense.*

Similarly to what is done for NTM, we can define the instantaneous descriptions of a NFTM $\mathscr{M} =$

[2] There are several notions of computability for functions defined in the interval [0, 1] (see [1,4,13,15,34]), however this will be irrrelevant for this paper.

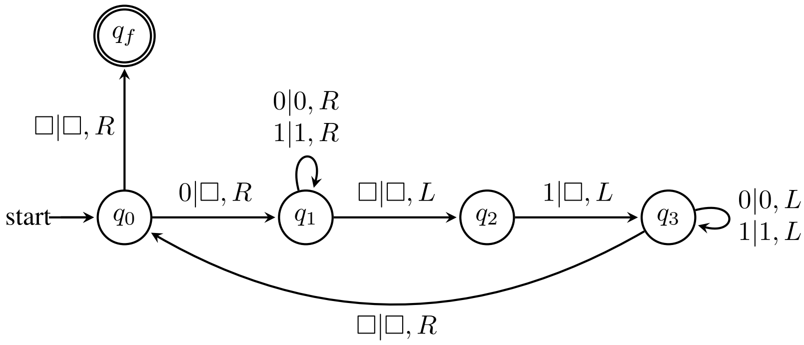

Fig. 2. Graphical representation of M

$\langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, \{\circledast_q\}_{q \in Q-F}, \mu\rangle$. An ID in $\mathscr{M}$ is a pair (uqv, r), where uvq is a classical ID in M and r is the degree of membership accumulated up to that point.

A valid movement between two ID's (D_1, r) and (D_2, s), denoted by $(D_1, r) \vDash_{\mathscr{M}} (D_2, s)$, occurs when there is a classical valid movement form $D_1 = uqv$ to $D_2 = u'q'v'$, with degree of truth t, and furthermore :

$$s = r \circledast_q t$$

Similarly to what we do for NTM, we extend recursively $\vDash_{\mathscr{M}}$, as follows:

Definition 10. *We say that* $(uqv, r) \vDash^*_{\mathscr{M}} (u'q'v', r')$ *if* $(uqv, r) = (u'q'v', r')$ *or is there a ID* (wpz, s) *such that* $(uqv, r) \vDash^*_{\mathscr{M}} (wpz, s) \vDash_{\mathscr{M}} (u'q'v', r')$.

Definition 11. *The acceptability of a string* w *in a NFTM* $\mathscr{M}$ *is given by*

$$deg_{\mathscr{M}}(w, k) = sup\{d \in [0,1] : (q_0 w, k) \vDash^*_{\mathscr{M}} (uq_f v, d) \text{ for some } q_f \in F \text{ and } u, v \in \Gamma^*\}.$$

When $k = 1$, *we write just* $deg_{\mathscr{M}}(w)$.

Note that $deg_{\mathscr{M}}$ associates each string $w \in \Sigma^*$ to one acceptability $deg_{\mathscr{M}}(w) \in [0,1]$, in other words $deg_{\mathscr{M}}$ can be interpreted as the membership function of a fuzzy subset of Σ^*. This allows us to define the fuzzy language accepted by a NFTM as a fuzzy set of strings, as follows:

$$\mathscr{L}(\mathscr{M}) = \{(w, deg_{\mathscr{M}}(w)) : w \in \Sigma^*\}$$

To make the above more general definition can use a t-conorm S extended to a countable number of real arguments, and define

$$deg^S_{\mathscr{M}}(w, k) = S(\{d \in [0,1] : (q_0 w, k) \vDash^*_{\mathscr{M}} (uq_f v, d) \text{ for some } q_f \in F \text{ and } u, v \in \Gamma^*\}),$$

and obviously define

$$\mathscr{L}^S(\mathscr{M}) = \{(w, deg^S_{\mathscr{W}}(w)) : w \in \Sigma^*\},$$

where $deg^S_{\mathscr{W}}(w) = deg^S_{\mathscr{W}}(w, 1)$.

Rewark 3. (i) *Note that when there is a unique* d *such that* $(q_0 w, 1) \vDash_{\mathscr{M}} (uq_f v, d)$, *by convention we have* $S(d) = d$, *and so* $deg^S_{\mathscr{W}}(w) = S(d) = d$.

(ii) *If does not exist* d *satisfying the above hypothesis, the set* $\{(w, deg^S_{\mathscr{W}}(w))) : w \in \Sigma^*\}$ *is empty, and thus* $deg^S_{\mathscr{W}}(w) = S(\emptyset) = 0$.

With all these preliminary notions, we can define the concept of fuzzy languages recursively enumerable *(LFRE)* in the following way:

Definition 12. *Let* $\mathscr{L} = \{w, \mu(w)) : w \in \Sigma^*\}$ *a fuzzy language. We say that* $\mathscr{L}$ *is a* **fuzzy recursively enumerable language**, *or* $\mathscr{L}$ *belongs to class LFRE, if there is a t-conorm* S *and a NFTM* $\mathscr{M}$ *such that* $\mathscr{L}^S(\mathscr{M}) = \mathscr{L}$.

Thus, a fuzzy Turing machine becomes of the form $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, \{\circledast_q\}, \mu, S\rangle$. Of course, all NTM can be seen as a NFTM according to the following proposition:

Proposition 2. *Let* $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, S\rangle$ *a Turing machine and* $\mathcal{L}$ *the accepted fuzzy language by* M. *Then, there is a NFTM* $\mathscr{M}$ *with fuzzy language* $\mathscr{L}(\mathscr{M}) = Crisp(\mathcal{L})$, *in other words,* $Crisp(\mathcal{L}) \in LFRE$.

Proof. Is enough to define $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, \{\circledast_q\}, \mu, S\rangle$, where $\circledast$ is any t-norm, $\mu(l) = 1$ (for all $l \in \delta$) and S is any t-conorm, for example S_{max}. $\square$

4. Operations between fuzzy Turing machines

In this section we will study the closure properties of fuzzy recursively enumerable languages, in general, show that the class *LFRE* if closed from operations: Union; Intersection; Dual and Reverse.

Theorem 1. *Let $\mathscr{L}_1 = \{(w, \mu_{\mathscr{L}_1}(w)) : w \in \Sigma^*\}$ and $\mathscr{L}_2 = \{(w, \mu_{\mathscr{L}_2}(w)) : w \in \Sigma^*\}$ two fuzzy languages on the same input alphabet. If there are NFTM's $\mathscr{M}_1 = \langle Q_1, \Sigma, \Gamma, \delta_1, q_0^1, \Box, F_1, \{\circledast_q^1\}_{q \in Q_1 - F_1}, \mu_1, S\rangle$ and $\mathscr{M}_2 = \langle Q_2, \Sigma, \Gamma, \delta_2, q_0^2, \Box, F_2, \{\circledast_q^2\}_{q \in Q_2 - F_2}, \mu_2, S\rangle$ with $\mathscr{L}(\mathscr{M}_1) = \mathscr{L}_1$ and $\mathscr{L}(\mathscr{M}_2) = \mathscr{L}_2$, then there is a NFTM $\mathscr{M}$ such that $\mathscr{L}(\mathscr{M}) = \mathscr{L}_1 \cup_S \mathscr{L}_2$.*

Proof. Without loss of generality, we can assume that $Q_1 \cap Q_1 = \emptyset$. Define the NFTM $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F, \{\circledast_q\}, \mu, S\rangle$ with $Q = q_0 \cup Q_1 \cup Q_2$, where $q_0 \notin Q_1 \cup Q_2$, $F = F_1 \cup F_2$, $\delta = \delta_1 \cup \delta_2 \cup \{(q_0, \sigma, q_0^i, \sigma, L) : \sigma \in \Gamma \text{ and } i = 1 \text{ or } 2\}$,

$$\mu(l) = \begin{cases} 1, & \text{if } l = (q_0, \sigma, q_0^i, \sigma, L) \text{ with } i = 1 \text{ or } 2 \\ \mu_1(l), & \text{if } l \in \delta_1 \\ \mu_2(l), & \text{if } l \in \delta_2 \end{cases},$$

for $l \in \delta$, and

$$x \circledast_q y = \begin{cases} xy, & \text{if } q = q_0 \\ x \circledast_q^1 y, & \text{if} q \in Q_1 \\ x \circledast_q^2 y, & \text{if } q \in Q_2 \end{cases}$$

With $(q_0 w, 1) \vDash_{\mathscr{M}} (q_0^1 w, 1)$ and $(q_0 w, 1) \vDash_{\mathscr{M}} (q_0^2 w, 1)$ for all $w \in \Sigma^*$. Thus, we have to

$$\begin{aligned} deg_{\mathscr{M}}^S(w) &= S(\{d \in [0,1] : (q_0 w, d) \vDash_{\mathscr{M}}^* (uq_f v, d) \\ &\qquad \text{for some } q_f \in F\}) \\ &= S(\{d \in [0,1] : (q_0^i w, d) \vDash_{\mathscr{M}}^* (uq_f v, d) \\ &\qquad \text{for some } q_f \in F_i \text{ and } i = 1 \text{ or } 2\}) \\ &= S(d_1, d_2), \end{aligned}$$

where $d_1 = S(\{d \in [0,1] : (q_0^1 w, d) \vDash_{\mathscr{M}}^* (uq_f v, d)$ for some $q_f \in F_1\})$ and $d_2 = S(\{d \in [0,1] : (q_0^2 w, d) \vDash_{\mathscr{M}}^* (uq_f v, d)$ for some $q_f \in F_2\})$. That is, $deg_{\mathscr{M}}(w) = S(deg_{\mathscr{M}_1}(w), deg_{\mathscr{M}_2}(w))$. However,

$$\begin{aligned} \mathscr{L}(\mathscr{M}) &= \{(w, deg_{\mathscr{M}}^S(w)) : w \in \Sigma^*\} \\ &= \{(w, S(deg_{\mathscr{M}_1(w)}, deg_{\mathscr{M}_1(w)}) : w \in \Sigma^*\} \\ &= \mathscr{L}_1 \cup_S \mathscr{L}_2 \\ &= \mathscr{L}(\mathscr{M}_1) \cup_S \mathscr{L}(\mathscr{M}_2) \end{aligned}$$

$\Box$

Example 4. *Consider the NTFM's $\mathscr{M}_1$ and $\mathscr{M}_2$, as represented in figure 3.*

The fuzzy languages accepted by $\mathscr{M}_1$ and $\mathscr{M}_2$, respectively are: $\mathscr{L}(\mathscr{M}_1) = \{(w, 0.2) : w \in \Sigma^*\}$ and $\mathscr{L}(\mathscr{M}_2) = \{(1, 0.4), (0, 0.1)\}$ (Rmk: All strings $w \in \Sigma^*$ with $|w| \geq 2$ are such that $deg_{\mathcal{M}_2}^S(w) = 0$). Easily, we can show that the NFTM represented by figure 4 recognizes the fuzzy language $\mathscr{L}(\mathscr{M}_1) \cup_S \mathscr{L}(\mathscr{M}_2)$ for any t-conorm S.

Rewark 4. *Note that the NFTM whose fuzzy language accepted corresponds to the union of two other languages ($\mathscr{L}(\mathscr{M}_1)$ and $\mathscr{L}(\mathscr{M}_2)$ in this case) is not unique, our construction merely ensures the existence of a NTFM computing the union of two fuzzy languages.*

Theorem 2. *Let $\mathscr{L} = \{(w, \mu_{\mathscr{L}}(w)) : w \in \Sigma^*\}$ a fuzzy language and $N : [0,1] \to [0,1]$ a strong negation. If there is a NFTM $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F, \{\circledast_q\}_{q \in Q - F}, \mu, S\rangle$ with $\mu(\delta) \subseteq (0,1)$ and such that $\mathscr{L}(\mathscr{M}) = \mathscr{L}$, then there is a NFTM $\mathscr{M}_N$ such that $\mathscr{L}(\mathscr{M}_N) = \mathscr{L}_N$.*

Proof. Let $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F, \{\circledast_q\}_{q \in Q - F}, \mu, S\rangle$ a NFTM such that $\mathscr{L}(\mathscr{M}) = \mathscr{L}$ and N a strong negation. Define $\mathscr{M}_N = \langle Q \cup \{\overline{q_0}\}, \Sigma, \Gamma, \delta', \overline{q_0}, \Box, F, \{\circledast_q'\}_{q \in Q - F} \cup \{\circledast_{\overline{q_0}}\}, \mu', S\rangle$, where $\delta' = \delta \cup \{(\overline{q_0}, \sigma, q_0, \sigma, L) : \sigma \in \Gamma\}$,

$$\mu'(l) = \begin{cases} N(\mu(l)), & \text{if } l \in \delta \\ 0.5, & \text{if } l = (\overline{q_0}, \sigma, q_0, \sigma, L) \end{cases}$$

and $\circledast_q' = \begin{cases} \circledast_{qN}, & \text{if } q \neq \overline{q_0} \\ \circledast_0, & \text{if } q = \overline{q_0} \end{cases}$, $l \in \delta'$, where $\circledast_0$ is a aggregation function such that $1 \circledast_0 0.5 = 0$ and $\circledast_{qN}$ is the dual of $\circledast_q$.

To prove that the accepted fuzzy language of NTFM given by $\mathscr{M}_N$ is $\mathscr{L}_N$ just check that

$$\begin{aligned} &(uqv, d) \vDash_{\mathscr{M}} (u'q'v', d') \iff \\ &\qquad (uqv, N(d)) \vDash_{\mathscr{M}_N} (u'q'v', N(d')). \end{aligned}$$

($\Longrightarrow$) If $(uqv, d) \vDash_{\mathscr{M}} (u'q'v', d')$, then by definition we have to

$$d' = d \circledast_q \mu(l),$$

where l corresponds to the transition that describes the movement from (uqv) to $(u'q'v')$. But since N is a strong negation, follows that

$$\begin{aligned} &N(d') = N(d \circledast_q \mu(l)) = \\ &\qquad N[N(N(d)) \circledast_q N(N(\mu(l)))] \end{aligned}$$

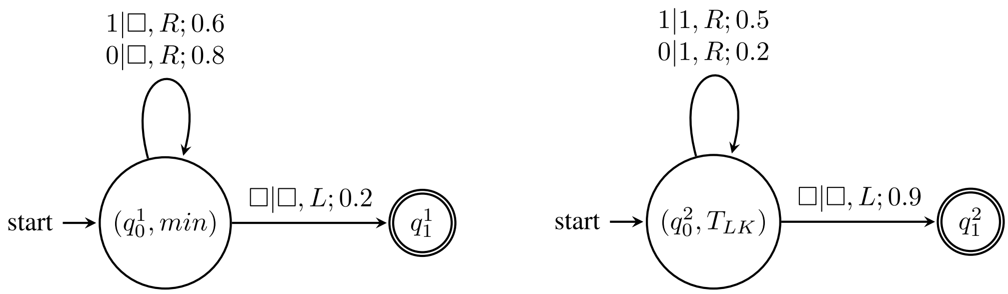

Fig. 3. Graphical representation of $\mathscr{M}_1$ and $\mathscr{M}_2$

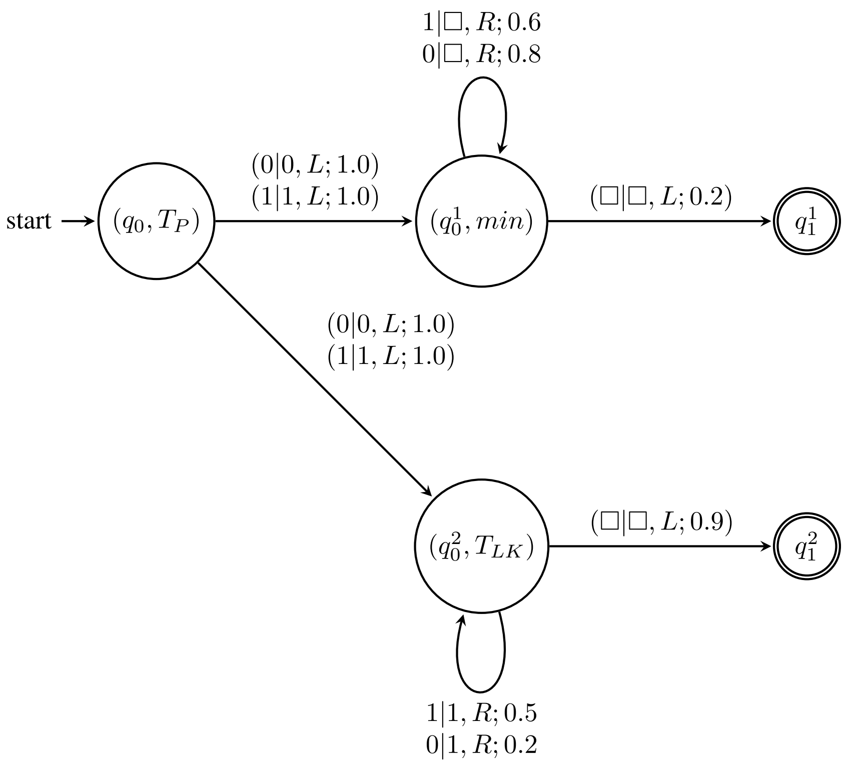

Fig. 4. Graphical representation of NFTM with fuzzy language $\mathscr{L}(\mathscr{M}_1) \cup_S \mathscr{L}(\mathscr{M}_2)$

$$= N(d) \circledast_{qN} N(\mu(l)).$$

Therefore,

$$(uqv, d) \vDash_{\mathscr{M}} (u'q'v', d') \Longrightarrow$$
$$(uqv, N(d)) \vDash_{\mathscr{M}_N} (u'q'v', N(d')).$$

($\Longleftarrow$) Following analogous to the previous case. $\square$

Example 5. *Consider the NTFM's of Example 3. Its dual with respect to the negation N_C are represented by figures 5 and 6:*

It is not difficult to show that the fuzzy language accepted by these machines are respectively $\mathscr{L}_{1N} = \{(w, 0.8) : w \in \Sigma^*\}$ and $\mathscr{L}_{1N} = \{(1, 0.6)\} \cup \{(0, 0.9)\}$. The aggregation function $\circledast_0$ can be defined, for example as

$$a \circledast_0 b = \begin{cases} 0, & \textit{if } a, b \in [0, 1) \\ 1, & \textit{otherwise} \end{cases}$$

4.1. Composition of fuzzy Turing machines

In this part of the paper we will present how we can perform composition of fuzzy Turing machine. This

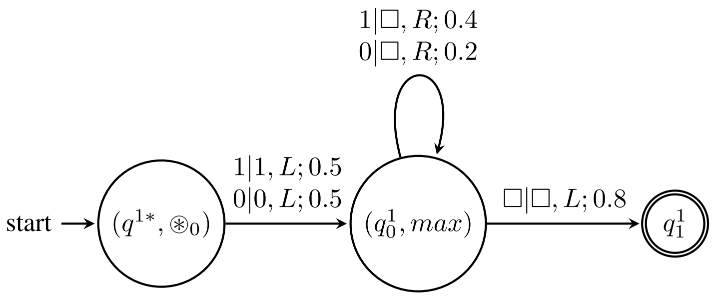

Fig. 5. Graphical representation of NFTM who accepts fuzzy language $\mathscr{L}(\mathscr{M}_1)_{N_C}$

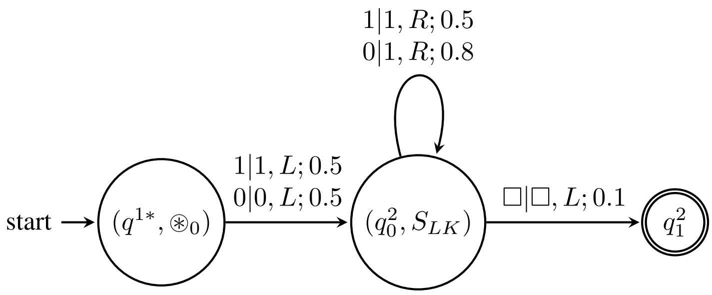

Fig. 6. Graphical representation of NFTM who accepts fuzzy language $\mathscr{L}(\mathscr{M}_2)_{N_C}$

tool will facilitate the proof of the results we present below.

Given two NFTM's $\mathscr{M}_1 = \langle Q_1, \Sigma, \Gamma, \delta_1, q_0^1, \square, F_1, \{\circledast_q\}_{q \in Q_1 - F_1}, \mu_1, S\rangle$ and $\mathscr{M}_2 = \langle Q_2, \Sigma', \Gamma', \delta_2, q_0^2, \square, F_2, \{\circledast_q\}_{q \in Q_2 - F_2}, \mu, S\rangle$ on the same input alphabet, where $\Gamma \subseteq \Sigma'$, we build a new NFTM, denoted by $\mathscr{M}_1 \circ \mathscr{M}_2 = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, \{\circledast_q\}_{q \in Q-F}, \mu, S\rangle$, where:

(i) We can assume without loss of generality, that $Q_1 \cap Q_2 = \emptyset$ and define $Q = Q_1 \cup Q_2$;

(ii) $q_0 = q_0^1$;

(iii) $F = F_2$;

(iv) (a) $\circledast_q = \begin{cases} \circledast_q, & \text{if } q \in Q_1 - F_1 \\ \circledast_q, & \text{if } q \in Q_2 - F_2 \end{cases}$;

(b) For each final state $q \in F_1$, define $\circledast_q$ so that $a \circledast_q 1 = a$;

(v) $\delta = \delta_1 \cup \delta_2 \cup \{(q, \sigma, q_0^2, \sigma, L) : q \in F_1 \text{ where } \sigma \in \Gamma'\}$;

(vi) $\mu(l) = \begin{cases} 1, & \text{if } l = (q, \sigma, q_0^2, \sigma, L) \text{ for some } q \in F_1 \\ \mu_1(l), & \text{if } l \in \delta_1 \\ \mu_2(l), & \text{if } l \in \delta_2 \end{cases}$.

In some applications it may be appropriate, after computing the first NFTM, return to the beginning of the working tape, and then initiate the computation of the second NTFN, while in other applications not.

This notion of composition to Turing machines will help us demonstrate the following results:

Proposition 3. *Let $\mathscr{L}$ a fuzzy language. If there is a NFTM $\mathscr{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F, \{\circledast_q\}_{q \in Q-F}, \mu, S\rangle$ such that $\mathscr{L}(\mathscr{M}) = \mathscr{L}$, then there is a NFTM $\mathscr{M}'$ such that $\mathscr{L}(\mathscr{M}') = \mathscr{L}^R$.*

Proof. With a little work we can build a classical Turing machine $\mathcal{M}_1$ which converts strings $w \in \Sigma^*$ in your reverse w^R, i.e, $q_0 w \vDash^*_{\mathcal{M}_1} q_f^1 w^R$. Futhermore, the proposition 2 ensures the existence of a NFTM $\mathscr{M}_1$ such that $(q_0 w, 1) \vDash^*_{\mathscr{M}_1} (q_f^1 w^R, 1)$ for all $w \in \Sigma^*$.

Thus, composing $\mathscr{M}_1$ with $\mathscr{M}$ we obtain a NFTM $\mathscr{M}_2 = \mathscr{M}_1 \circ \mathscr{M}$ such that $\mathscr{L}(\mathscr{M}_2) = \mathscr{L}^R$, since for all $w \in \Sigma^*$ with $(q_0 w, 1) \vDash^*_{\mathscr{M}} (uq_f v, d)$, accepted by $\mathscr{M}$, we have to:

$$(q_0 w^R, 1) \vDash^*_{\mathscr{M}_1} (q_f^1 w, 1) \vDash_{\mathscr{M}} (q_0 w, 1) \vDash^*_{\mathscr{M}} (uq_f v, d)$$

This shows that

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (uq_f v, d) \Longrightarrow (q_0^1 w^R, 1) \vDash^*_{\mathscr{M}_2} (uq_f v, d)$$

The reciprocal of the above assertion follows building of $\mathscr{M}_2 = \mathscr{M}_1 \circ \mathscr{M}$. $\square$

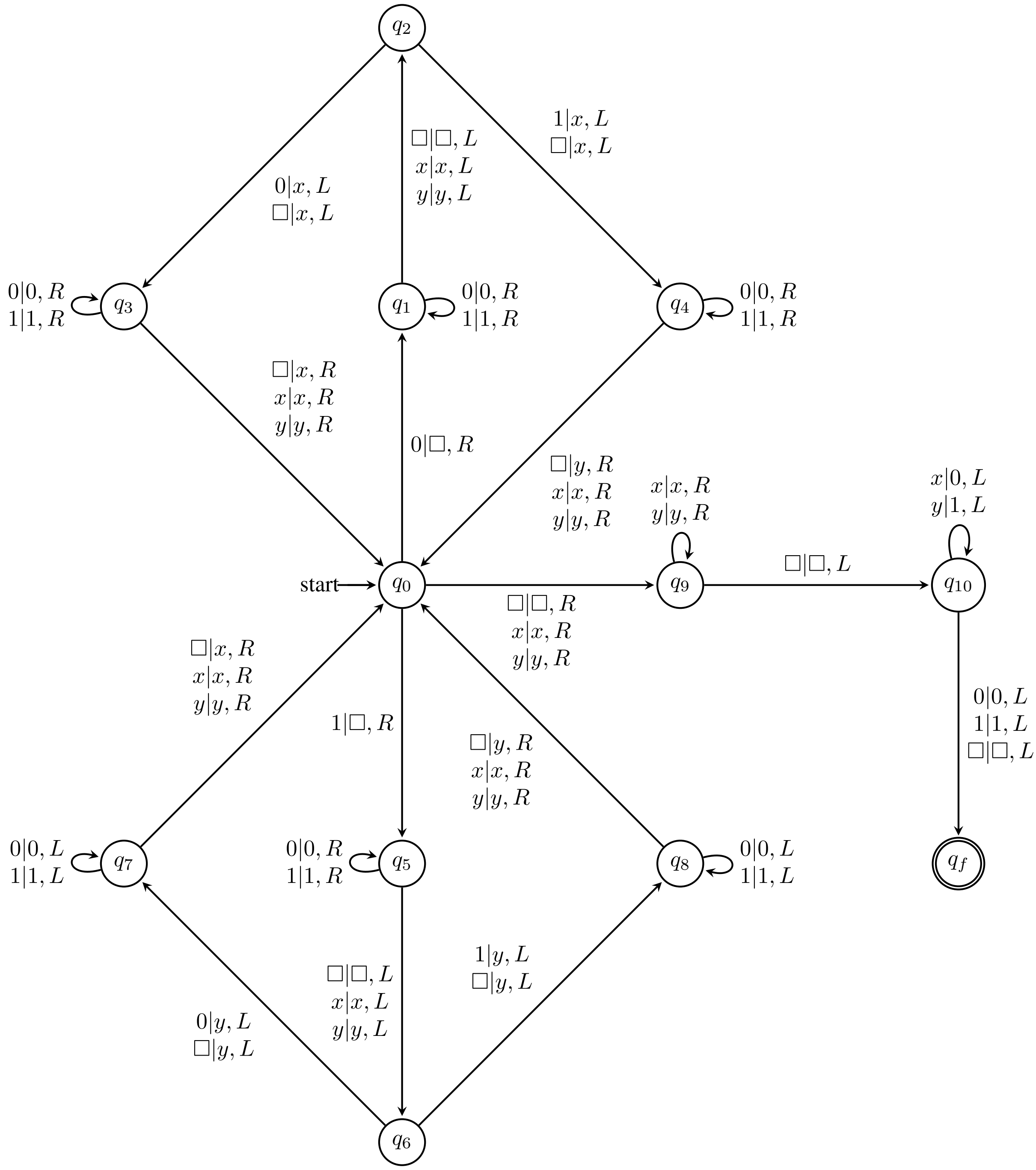

Fig. 7. Graphical representation of a classical Turing machine that reverses any string $w \in \{0, 1\}^*$

Example 6. *To illustrate the process of reversal of strings, we present, in figure 7, the classical diagram for a Turing machine which reverses any* $w \in \{0, 1\}^*$.

So far we have proven constructions that allow us to operate on unions, dual and reverse of recursively enumerable languages by NFTM's. We are now interested in building fuzzy Turing machine able to compute the intersection of two recursively enumerable languages.

Given a family $\mathcal{F}$ nor empty of aggregation functions, we say that $\mathcal{F}$ satisfies the property of **generalized**

associativity (see [27]) if for any pair of aggregations functions $A, B \in \mathcal{F}$ we have

$$A(x, B(y,z)) = B(A(x,y), z), \text{ for all } x, y, z \in [0,1]$$

Obviously, given any t-norm T, the family $\mathcal{F} = \{T\}$ satisfies the property of generalized associativity. In general, given any aggregation function A, the family $\mathcal{F} = \{A\}$ satisfies the property of generalized associativity if, and only if, A is associative aggregation function.

In the following proposition, we show a infinite family of aggregation functions that satisfies the generalized associativity.

Proposition 4. *Let $\mathcal{F} = \{T_\alpha(x,y) : \alpha \geq 1\}$, where*

$$T_\alpha(x,y) = \begin{cases} 1, & \text{if } x \cdot y = 1 \\ \frac{xy}{\alpha}, & \text{otherwise} \end{cases}.$$

Then, $\mathcal{F}$ satisfies the generalized associativity.

Proof. Obviously, for each $\alpha \geq 1$ fixed we have to T_α is a aggregation function. Now, given $\alpha, \beta \geq 1$:

$$T_\alpha(x, T_\beta(y,z)) = \begin{cases} 1, & \text{if } x \cdot T_\beta(y,z) = 1 \\ \frac{x \cdot T_\beta(y,z)}{\alpha}, & \text{otherwise} \end{cases}$$

In the case that $x \cdot T_\beta(y,z) = 1$ we have

$$x = 1 \text{ and } T_\beta(y,z) = 1 \Longrightarrow x = y = z = 1$$
$$\Longrightarrow T_\alpha(x, T_\beta(y,z)) = T_\beta(T_\alpha(x,y), z) = 1.$$

In the other case,

$$T_\alpha(x, T_\beta(y,z)) = \begin{cases} \frac{x}{\alpha}, & \text{if } T_\beta(y,z) = 1 \\ \frac{xyz}{\alpha\beta}, & \text{otherwise} \end{cases}$$
$$\Longrightarrow T_\alpha(x, T_\beta(y,z)) = T_\beta(T_\alpha(x,y), z).$$

□

We can not always guarantee that the intersection of recursively enumerable fuzzy languages results in a recursively enumerable fuzzy language. The following theorem guarantees us sufficient conditions for this to occur.

Theorem 3. *Let $\mathscr{L}_1$ and $\mathscr{L}_2$ two fuzzy languages. If there is a NFTM's $\mathscr{M}_1 = \langle Q_1, \Sigma, \Gamma, \delta_1, q_0^1, \Box, F_1, \{\circledast_q\}_{q \in Q_1 - F_1}, \mu_1, S\rangle$ and $\mathscr{M}_2 = \langle Q_2, \Sigma, \Gamma, \delta_2, q_0^2, \Box, F_2, \{\circledast_q\}_{q \in Q_2 - F_2}, \mu_2, S\rangle$ such that $\mathscr{L}(\mathscr{M}_1) = \mathscr{L}_1$, $\mathscr{L}(\mathscr{M}_2) = \mathscr{L}_2$, $\{\circledast_q\}_{q \in (Q_1 \cup Q_2) - (F_1 \cup F_2)}$ is a subset of family of aggregation functions $\mathcal{F}$ satisfying the generalized associativity and $T \in \mathcal{F}$ is a t-norm t-norma such that $S(\{T(x_i, y_i)\}) = T(S(X), S(Y))$ for any countable sequence $X = \{x_1, x_2, \cdots\}$ and $Y = \{y_1, y_2, \cdots\}$, then there is a NFTM $\mathscr{M}$ with $\mathscr{L}(\mathscr{M}) = \mathscr{L}_1 \cap_T \mathscr{L}_2$.*

Proof. Without loss of generality assume that $Q_1 \cap Q_1 = \emptyset$. The proof of this theorem will be held the following steps:

Step 1: Add to input alphabets and output a special symbol denoted by $\#$, and redefine $\mathscr{M}_1$ and $\mathscr{M}_2$ for these new alphabets keeping the transitions δ_1 and δ_2, adding to the two machines the transitions of the form $(q, \#, q, \#, R)$ for every state q.

Step 2: Let $\mathscr{M}_0$ a crisp NFTM that performs the following operation:

$$(q_0^0 w, 1) \vDash^*_{\mathscr{M}_0} (w q_f^0 \# w, 1) \text{ for all } w \in \Sigma^*$$

Where all aggregate functions Machine $\mathscr{M}_0$ belonging to the family $\mathcal{F}$.

Step 3: Define another crisp NFTM $\mathscr{M}_3$ with only one aggregation function, the t-norm T_1, such that:

$$(w q_0^3 \# v, 1) \vDash^*_{\mathscr{M}_3} (q_f^3 w, 1) \text{ for all } w, v \in \Sigma^*$$

Step 4: Build a NFTM $\mathscr{M}$ from the composition

$$\mathscr{M}_1 \circ (\mathscr{M}_3 \circ (\mathscr{M}_2 \circ \mathscr{M}_0))$$

using the following criteria:

(a) The transitions of q_f^0 to q_0^2 have a form $(q_f^0, \#, q_0^2, \#, R)$ with degree of truth equal to 1;
(b) Add the final states of $\mathscr{M}_2$ the transitions $(q_f^2, \sigma, q_f^2, \sigma, L)$ and $(q_f^2, \#, q_0^3, L)$ with degree of truth equal to 1;
(c) Associate to the states q_f^3 the t-norm T_1;
(d) The transitions of q_f^3 to q_0^1 have a form (q_f^3, w, q_0^1, w, L) with degree of truth equal to 1.

Step 5: For any $w \in \Sigma^*$ we have

$$(q_0^0 w, 1) \vDash^*_{\mathscr{M}_0} (w q_f^0 \# w, 1) \Longrightarrow$$
$$(q_0^0 w, 1) \vDash^*_{\mathscr{M}} (w q_f^0 \# w, 1) \vDash_{\mathscr{M}} (w \# q_0^2 w, 1)$$
$$\Longrightarrow (q_0^0 w, 1) \vDash^*_{\mathscr{M}} (w \# q_0^2 w, 1)$$

Now, if w is such that $(q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (uq_f^2 v, d)$, then

$$(q_0^0 w, 1) \vDash^*_{\mathscr{M}} (w\#q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (w\#uq_f^2 v, d) \Longrightarrow (q_0^0 w, 1) \vDash^*_{\mathscr{M}} (w\#uq_f^2 v, d)$$

By (b) we obtain that

$$(w\#uq_f^2 v, d) \vDash^*_{\mathscr{M}} (wq_f^2\#uv, d)$$

At this point we enter the computing of machine $\mathscr{M}_3$, whose final result is $(q_f^3 w, d)$. That is,

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (q_f^3 w, d)$$

for all $w \in \Sigma^*$ with $(q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (uq_f^2 v, d)$. But, by (c) and (d) we have to

$$(q_f^3 w, d) \vDash_{\mathscr{M}} (q_0^1 w, T(d, 1))$$

Therefore,

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (q_0^1 w, T(d, 1))$$

for all $w \in \Sigma^*$ with $(q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (uq_f^2 v, d)$. Now suppose that $(q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (uq_f^2 v, d)$ and $(q_f^1 w, 1) \vDash^*_{\mathscr{M}_1} (uq_f^1 v, d')$. Then, there is a finite sequÃłnce of ID's in $\mathscr{M}_1$ such that:

$$(q_0^1 w, 1) = (D_1, 1) \vDash_{\mathscr{M}_1} (D_2, r_2) \vDash_{\mathscr{M}_1} \cdots \vDash_{\mathscr{M}_1} (D_{n-1}, r_{n-1}) \vDash D_n = (uq_f^1 v, d')$$

Let l_i the occurred transition between the ID's D_i and D_{i+1}, for $i \in \{1, \cdots, n-1\}$, and $\mu_1(l_i)$ the degrees of truth of these transitions. Then, with $(q_0^2 w, 1) \vDash^*_{\mathscr{M}_2} (uq_f^2 v, d)$ we have to

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (q_f^3 w, d) \vDash_{\mathscr{M}} (q_0^1 w, T(d, 1)) \vDash_{\mathscr{M}} (D_1, T(d, 1) \circledast_{q_0^1} \mu_1(l_1)),$$

but how T_1 and $\circledast_{q_0^1}$ satisfy the generalized associativity, we observe that

$$T(d, 1) \circledast_{q_0^1} \mu_1(l_1)) = T(d, 1 \circledast_{q_0^1} \mu_1(l_1)) \Longrightarrow (q_0 w, 1) \vDash^*_{\mathscr{M}} (D_1, T(d, 1 \circledast_{q_0^1} \mu_1(l_1))).$$

Similarly we have that, if

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (D_i, T(d, (((1 \circledast_{q_0^1} \mu_1(l_1)) \circledast_{q_{k_1}^1}) \cdots) \circledast_{q_{k_{i-1}}^1})),$$

then

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (D_{i+1}, T(d, (((1 \circledast_{q_0^1} \mu_1(l_1)) \circledast_{q_{k_1}^1}) \cdots) \circledast_{q_{k_i}^1})).$$

Therefore,

$$(q_0 w, 1) \vDash^*_{\mathscr{M}} (D_n, T(d, (((1 \circledast_{q_0^1} \mu_1(l_1)) \circledast_{q_{k_1}^1}) \cdots) \circledast_{q_{k_{n-1}}^1})) = (uq_f^1 v, T(d, d')).$$

Thus, we conclude that

$$deg^S_{\mathscr{M}}(w) = T(deg^S_{\mathscr{M}_2}(w), deg^S_{\mathscr{M}_1}(w)).$$

□

5. Final remarks

In this paper we proposed a model of non-deterministic fuzzy Turing machine - NFTM, which generalizes the models available in the literature. Usually a fuzzy Turing machine $\mathscr{M}$ comes equipped with a connective operator, a t-norm, so that you can compute the degree of acceptance of a string and thus determine the fuzzy language accepted by $\mathscr{M}$. However, we realized that t-norms have unnecessary characteristics such as commutativity and associativity, so we replace this unique operator for various aggregation functions. In addition, we established the fuzzy language computable, or fuzzy recursively enumerable languages - *LFRE*, according to our model, and we prove, among other properties, that the class of languages *LFRE* is closed for operations such as: Unions and widespread intersections; Reverse and Dual.

References

[1] B. C. Bedregal, B.M. Acióly, A computability on the interval space: a domain approach, in: Proc. XXIII Conf. Latinoamericana de Informática, v. I, Valparaíso-Chile (1997), pp. 107-116.

[2] B. C. Bedregal, S, Figueira, On the computing power of fuzzy Turing machines, Fuzzy Sets and Systems, v. 159 (2008), 1072-1083.

[3] G. Beliakov, A. Pradera, T. Calvo, Aggregation functions: a guide for practitioners, Stud. Fuzziness Soft Comput, 2007.

[4] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bull. Amer. Math. Soc. v. 21(1) (1989), pp. 1-46.

[5] L. Biacino, G. Gerla, Fuzzy subsets: a constructive approach, Fuzzy Sets and Systems, v. 45 (1992), pp. 161-168.

[6] L. Biacino, G. Gerla, Fuzzy logic, continuity and effectiveness, Arch. Math. Logic, v. 41 (2002), pp. 643-667.

[7] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, B. Bedregal, J. Montero, H. Hagras, F. Herrera, B. De Baets, A historical account of types of fuzzy sets and their relationships, IEEE Transactions on Fuzzy Systems (2015).

[8] B. Clares, M. Delgado, Introduction of the concept of recursiveness of fuzzy functions, Fuzzy Sets and Systems v. 21 (1987), pp. 301-310.

[9] A.C.R. Costa, G.P. Dimuro, Interactive computation: stepping stone in the pathway from classical to developmental computation, Electronic Notes in Theoretical Computer Science, v. 141(5) (2005), pp. 5-31.

[10] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, Proc. Roy. Soc. London Ser., v 400 (1985), pp. 97-117.

[11] G. Gerla, Fuzzy Logic: Mathematical Tools for Approximate Reasoning, Kluwer Academic Publishers, Dordrecht, 2001.

[12] G. Gerla, Effectiveness and multivalued logics, J. Symbolic Logic, v. 71 (2006), pp. 137-162.

[13] P. Di Gianantonio, Real number computation and domain theory, Inform. and Comput., v. 127 (1996), 11-25.

[14] D.Q. Goldin, S.A. Smolka, P.C. Attie, E.L. Sonderegger, Turing machines, transition systems and interaction, Inform. and Comput., v. 194 (2004), pp. 101-128.

[15] A. Grzegorczyk, On the definitions of computable real continuous functions, Fund. Math., v. 44 (1957), pp. 61-71.

[16] L. Harkleroad, Fuzzy recursion, ret's and isols, Z. Math. Logik Grundlagen Math., v. 30 (1984), pp. 425-436.

[17] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, Reading, MA, 1979.

[18] E. P. Klement, R. Mesiar, E. Pap, Triangular Norms, Springer, v. 8, 2000.

[19] E.T. Lee, L.A. Zadeh, Note on fuzzy languages, Inform. Sci., v. 4(1) (1969), pp. 421-434.

[20] Y. Li, Fuzzy Turing Machines: Variants and Universality, IEEE Trans. on Fuzzy Systems, v. 16, n. 6 (2008), pp. 1491-1502.

[21] Y. Li, Lattice-valued fuzzy Turing machines: Computing power universality and efficency, Fuzzy Sets and Systems, v. 160 (2009), pp. 3453-3474.

[22] P. Linz, An Introduction to Formal Language and Automata, Jones and Bartlett Publisher, 2001.

[23] K. Menger, Statistical metrics. Proc. Nat. Acad. Sci. USA, v. 28 (1942), pp. 535-537.

[24] C. Moraga, Towards a fuzzy computability, Mathware & Soft Comput., v. 6 (1999), pp. 163-172.

[25] R. Morales-Bueno, R. Conejo, J.L. Pérez de-la Cruz, F. Triguero-Ruiz, On a class of fuzzy computable functions, Fuzzy Sets and Systems, v. 121 (2001), pp. 505-522.

[26] J.N. Morderson, D.S. Malik, Fuzzy Automata and Languages: Theory and Applications, Chapman & Hall, CRC, London, Boca Raton, FL, 2002.

[27] R. H. S. Reiser,, B. Bedregal, M. Baczynski, Aggregating fuzzy implications, Inform. Sci., v. 253 (2013), pp. 126-146.

[28] A. Turing, On computable numbers, with an application to the Entscheidungs problem, Proc. Lond. Math. Soc., v. 42, no. 2 (1936), pp. 230-260.

[29] A. Turing, On computable numbers, with an application to the Entscheidungs problem, Proc. Lond. Math. Soc., v. 43, no. 2 (1937), pp. 544-546.

[30] E.S. Santos, Fuzzy algorithms, Inform. and Control, v. 17 (1970), pp. 326-339.

[31] E.S. Santos, Fuzzy and probabilistic programs, Inform. Sci., v. 10 (1976), pp. 331-335.

[32] B. Schweizer, A. Sklar, Associative functions and statistical triangle inequalities. Publ. Math. Debrecen. v. 8 (1961), pp. 169-186.

[33] B. Schweizer, A. Sklar, Probabilistic metric spaces. North-Holland, New York, 1983.

[34] K. Weihrauch, Computable Analysis: An Introduction, Springer, Berlin, 2000.

[35] J. Wiedermann, Fuzzy Turing machines revised, Comput. Artificial Intelligence, v. 21(3) (2002), pp. 1-13.

[36] J. Wiedermann, Characterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines, Theoret. Comput. Sci., v. 317 (2004), pp. 61-69.

[37] L. Wu, D. Qiu, H. Xing, Automata theory based on complete residuated lattice-valued logic: Turing machines, Fuzzy Sets and Systems, v. 208 (2012), pp. 43-66.

[38] L. A. Zadeh, Fuzzy Sets, Inform. Control, v. 8 (1965), pp. 338-353.

[39] L. A. Zadeh, Fuzzy algorithms, Inform. and Control, v. 2 (1968), pp. 94-102.